Structure-Adaptive Accelerated Coordinate Descent

Junqi Tang, Mohammad Golbabaee, Francis Bach, Mike E. Davies

Structure-Adaptive Accelerated Coordinate Descent

Junqi Tang University of Edinburgh
Mohammad Golbabaee University of Bath
Francis Bach INRIA, Paris
Mike Davies University of Edinburgh

Abstract

In this work we explore the fundamental structure-adaptiveness of accelerated randomized coordinate descent algorithms on regularized empirical risk minimization tasks, where the solution has intrinsic low-dimensional structure such as sparsity and low-rank, enforced by non-smooth regularization. We propose and analyze a two-stage accelerated coordinate descent algorithm ("two-stage APCG") utilizing the restricted strong-convexity framework. We provide the convergence analysis showing that the proposed method have a local accelerated linear convergence rate with respect to the low-dimensional structure of the solution. We also propose an adaptive variant of the two-stage APCG which does not need to foreknow the restricted strong convexity parameter beforehand, but estimates it on the fly. In our numerical experiments we test the proposed method on a number of machine learning datasets and demonstrate the effectiveness of our approach.

1 INTRODUCTION

Many applications in machine learning, signal processing and computer vision share the same goal, which is to achieve a good estimation of the minimizer $x^\dagger \in \mathbb{R}^m$ of the expected risk function: $x^\dagger = \arg\min_x \mathbb{E}\bar{f}(x)$ via minimizing the empirical risk $f(x)$, (Vapnik, 2013). In machine learning practice, the number of training data is usually limited and the parameter space can be very high-dimensional, hence minimizing the empirical risk $f(x)$ alone will introduce overfitting and fails to get a reasonable estimation of $x^\dagger$ (Wainwright, 2014; Bühlmann and Van De Geer, 2011; Tibshirani et al., 2015). To avoid this, a standard approach is to introduce regularization in additional to the empirical risk (Bickel et al., 2006; Bach et al., 2012). We thus consider the convex composite minimization task which reads:

$$x^\star \in \arg\min_{x\in\mathbb{R}^m} \left\{F(x) := f(x) + \lambda g(x)\right\}, \qquad (1)$$

where x consists of d-blocks of subvectors: $[x_{(1)}, ..., x_{(d)}]$ and the regularization term $g(x)$ is potentially non-smooth but separable such that $g(x) = \sum_{i=1}^d g_i(x_{(i)})$, and $f(x)$ is differentiable with Lipschitz-continuous gradients. When the minimization task is large-scale and high-dimensional, the traditional deterministic gradient methods typically fail to achieve scalablilty. To address this, randomized coordinate descent (RCD) (Nesterov, 2012; Richtárik and Takáč, 2014; Lu and Xiao, 2015) has been intensely studied and widely applied due to its efficiency in solving many types of high-dimensional problems (Hsieh et al., 2008; Wu et al., 2008; Wen et al., 2012; Qin et al., 2013). To further improve the convergence speed of the coordinate descent method, researchers have successfully combined it with Nesterov's acceleration technique (Nesterov, 1983, 2007, 2013), and developed *accelerated coordinate descent* algorithms (Nesterov, 2012; Lee and Sidford, 2013; Fercoq and Richtárik, 2015; Lin et al., 2014, 2015) which enjoy optimal worst-case convergence speed in theory, and much improved practical performance over vanilla coordinate descent. Very recently researchers have even proposed several successful variants of accelerated coordinate descent which are based on various schemes such as restart (Fercoq and Qu, 2016, 2018), non-uniform sampling (Allen-Zhu et al., 2016; Nesterov and Stich, 2017) and Gauss-Southwell greedy selection rules (Lu et al., 2018).

1.1 The Solution's Structure and Faster Convergence

While researchers have developed the so-called optimal coordinate descent algorithms for the composite optimization tasks (1), these algorithms do not take advantage of the prior information brought forth by

Technical Report. Part of the results of this paper occurred in our unpublished earlier version (Tang et al., 2017).

the regularization term $g(x)$. Popular non-smooth regularization applied in machine learning and signal processing applications enforce the solution to have low-dimensional structure, for example the sparsity, group-sparsity or low-rank. In this work, by introducing a simple variant of the *accelerated proximal coordinate gradient* (APCG) algorithm of Lin et al. (2014), we show that one can significantly improve the convergence speed of these methods if the prior information is properly exploited.

One key theoretical milestone in work on structure-adaptive convergence is the *modified restricted strong-convexity* framework developed by Agarwal et al. (2012). Their work provides the first result on the linear convergence speed of the proximal gradient descent method in the context of high-dimensional statistical estimation, where the standard strong-convexity assumption is vacuous. However, in such a setting restricted strong-convexity may still hold true thanks to the low-dimensional structure of the solution promoted by the regularization. More inspiringly, their result indicates that the fewer degrees of freedom (in other words, more structured) the solution has, the faster the convergence. This result confirms the intuition that a truly optimal coordinate descent method for (1) should be able to exploit the prior information given by the regularizer, and the solution's low-dimensional structure. Furthermore, very recently, researchers extended this framework to analyze the stochastic variance-reduced gradient (SVRG) methods (Qu and Xu, 2016, 2017), and proposed an accelerated variant of it which is provably able to exploit the solution's structure for even faster convergence (Tang et al., 2018).

1.2 This Work

In this paper, we make the following contributions.

Theoretical Contributions. We analyze the relationship between the solution's low-dimensional structure and the convergence speed of accelerated coordinate descent methods in the primal form. We choose to use the accelerated proximal coordinate descent method APCG (Lin et al., 2014, 2015) as the foundation to build up our novel "Two-Stage APCG" method which is dedicated to actively exploit the intrinsic low-dimension structure of the solution prompted by the (non-smooth) regularization. The convergence analysis shows that the Two-Stage APCG method exhibits global convergence: in the first stage, the method converges sublinearly to the vicinity of the solution, while in the second stage the method converges towards the solution with an accelerated linear rate with respect to the modified restricted strong convexity (RSC) (Agarwal et al., 2012) which scales with the solution's intrinsic dimension.

Algorithmic Contributions. We propose an adaptive restart variant of our Two-Stage APCG algorithm which is motivated by our underlying theory and does not need explicit knowledge of the restricted strong convexity (RSC) parameter but still provides excellent practical performance. In practice the strong convexity and also restricted strong convexity parameter cannot be easily obtained beforehand in general practical setups, which is necessary for the accelerated methods to achieve accelerated linear convergence rate (Nesterov, 2012; Wright, 2015; Arjevani, 2017). To overcome this issue we propose an adaptive variant of the two-stage APCG method which is based on a simple heuristic scheme to estimate the RSC on the fly. Tested on a number of high-dimensional datasets, our experiments demonstrate the effectiveness of our algorithm.

2 TWO-STAGE APCG

In this section we start by introducing the vanilla accelerated coordinate method APCG developed by Lin et al. (2015), and then our Two-Stage APCG method which has the desirable structure-adaptive property.

We first list some standard notations following the accelerated coordinate descent literature (Fercoq and Richtárik, 2015; Lin et al., 2014, 2015).

Definition 2.1. *(Block Coordinate Structure and Partial Gradients.) We split the full space $\mathbb{R}^m$ into d blocks of subspaces, that is, for any vector $x \in \mathbb{R}^m$ with $\left\{x_{(i)} \in \mathbb{R}^{m_i}, \quad i = 1, ..., d, \quad \sum_i m_i = m\right\}$, there is a permutation matrix $U \in \mathbb{R}^{m\times m}$ with submatrices $\left\{U = [U_1, ..., U_d], U_i \in \mathbb{R}^{m\times m_i}, i = 1, ..., d\right\}$ such that $x = \sum_{i=1}^d U_i x_{(i)}$. We also define the partial gradient of the smooth function $f(.)$ w.r.t $x_{(i)}$ as:*

$$\nabla_i f(x) = U_i^T \nabla f(x). \tag{2}$$

Moreover the regularization term has block-coordinate-wise separable structure: $g(x) = \sum_{i=1}^d g_i(x_{(i)})$.

We assume that $f(.)$ has block-coordinate-wise Lipschitz continuous gradient with parameter L_i for each block of coordinates $i \in [1, d]$, and define a weighted norm $\|x\|_L = \left(\sum_{i=1}^d L_i \|x_{(i)}\|_2^2\right)^{1/2}$. We list the details of the APCG algorithm (Lin et al., 2015, Alg. 2) for strongly-convex functions:

APCG(x_0, K, α)**:**

For $k = 0, 1, 2, ..., K$

$$\begin{aligned}
&y_k = \tfrac{x_k + \alpha z_k}{1+\alpha};\\
&z_{k+1} = \arg\min_{x\in\mathbb{R}^d} \tfrac{\alpha d}{2}\|x - (1-\alpha)z_k - \alpha y_k\|_L^2\\
&\qquad\qquad + \langle \nabla_{i_k} f(y_k), x_{(i_k)}\rangle + \lambda g_{i_k}(x_{(i_k)})\\
&x_{k+1} = y_k + d\alpha(z_{k+1} - z_k) + d\alpha^2(z_k - y_k);
\end{aligned}$$

with initialization $z_0 = x_0$, $\alpha = \frac{\sqrt{\mu}}{d}$, and in each iteration an index $i_k \in 1, ..., d$ is chosen uniformly at random, and we take the result of the last iterate (x_{K+1}) as the output. If the objective function F is strongly-convex, then the APCG algorithm enjoys a Nesterov-type accelerated linear convergence rate. Similarly we also provide the details of the APCG algorithm for minimizing non-strongly-convex functions (Lin et al., 2015, Alg. 3), which we denote as APCG_0, with initialization $z_0 = x_0$ and $\alpha_{-1} = \frac{1}{d}$:

$\text{APCG}_0(x_0, K)$:

For $k = 0, 1, 2, ..., K$

$$\begin{aligned}
&\alpha_k = \tfrac{1}{2}(\sqrt{\alpha_{k-1}^4 + 4\alpha_{k-1}^2} - \alpha_{k-1}^2),\\
&y_k = (1-\alpha_k)x_k + \alpha_k z_k.\\
&z_{k+1} = \arg\min_{x\in\mathbb{R}^d} \tfrac{\alpha d}{2}\|x - z_k\|_L^2\\
&\qquad\qquad +\langle \nabla_{i_k} f(y_k), x_{(i_k)}\rangle + \lambda g_{i_k}(x_{(i_k)})\\
&x_{k+1} = y_k + d\alpha(z_{k+1} - z_k);
\end{aligned}$$

If the objective function is convex but non-strongly-convex, the APCG_0 has an $O(1/k^2)$ accelerated sublinear convergence rate. These convergence rates match the optimal worse-case rates of Nesterov's accelerated gradient method (Nesterov, 2007) for $d = 1$ and improve upon the proximal coordinate descent (Richtárik and Takáč, 2014) for $d > 1$. However, in many high-dimensional applications the strong-convexity assumption is vacuous, while the non-strongly-convex assumption is too weak with structure-promoting regularization. As shown by Agarwal et al. (2012), with a sufficient amount of non-smooth structure-promoting regularization such as ℓ_1 norm, $\ell_{1,2}$ norm, or nuclear norm penalty, the objective function is "strongly-convex" locally around the solution from a restricted range of directions. This phenomenon is characterized as the *restricted strong-convexity* (RSC). The APCG algorithm itself cannot directly exploit the RSC to achieve faster convergence in theory.

To exploit the structure of the solution for faster convergence, we propose variants of accelerated coordinate descent algorithms base on the APCG, under a two-stage splitting framework inspired by the local nature of the RSC: at the first stage for warm-starting, we run the non-strongly-convex APCG_0 algorithm to a neighborhood of the solution; at the second stage, since a local linear convergence rate is expected due to the RSC, we have two choices: (1) periodically restart the non-strongly-convex APCG_0 at a certain frequency w.r.t the RSC parameter μ_c, which leads to our Option 1, (2) run the APCG algorithm with the momentum parameter $\alpha = \frac{\sqrt{\mu_c}}{d}$ and a restart period also w.r.t μ_c, which leads to Option 2. We describe the two-stage APCG as Algorithm 1, where we use superscript t to index outer-loop and subscript k to index inner-loop of our algorithms.

Algorithm 1 Two-Stage APCG

Inputs: x^0 and restricted strong-convexity parameter μ_c, number of iteration K_0 for the first stage; $T \geq 1$; $\beta \geq 2$

1. First stage, start without μ_c:

$$x^1 = \text{APCG}_0(x^0, K_0) \tag{3}$$

2. Second stage – exploit local accelerated linear convergence given by μ_c

Option 1: with $K = \left\lceil 2d\beta\sqrt{2 + \frac{1}{\mu_c}} - 2d \right\rceil$

for $t = 1, \ldots, T$ **do**

$$x^{t+1} = \text{APCG}_0(x^t, K) \tag{4}$$

end for

Output: x^{T+1}

Option 2: with $K = \left\lceil \frac{\log 16}{\log \frac{1}{1-\sqrt{\mu_c}/d}} \right\rceil$

for $t = 1, \ldots, T$ **do**

$$x^{t+1} = \text{APCG}(x^t, K, \mu_c) \tag{5}$$

end for

Output: x^{T+1}

We need to point out that our algorithm with Option 1 is a two-stage variant of the Restarted-APPROX algorithm of Fercoq and Qu (2016) which is also based on restarting the accelerated coordinate descent. This algorithm was originally designed for minimizing functions which satisfy a quadratic error bound condition – a condition which is also weaker than strong-convexity but does not encode the solution's structure enforced by regularization. The Restarted-APPROX algorithm on its own does not have theoretical convergence result under the RSC framework of Agarwal et al. (2012) which is relevant to the purpose of this work.

2.1 Generic Assumptions

In this section we list out the assumptions which we required in our convergence proofs. Similar assumptions have been used in the related literature (Agarwal et al., 2012; Qu and Xu, 2016; Tang et al., 2018).

A. 1. *(Block-Coordinate Smoothness.) Assume that $f(x)$ has block-coordinate-wise Lipschitz continuous gradient:*

$$\|\nabla_i f(x + U_i h_i) - \nabla_i f(x)\|_2 \leq L_i \|h_i\|_2, \tag{6}$$

$\forall h_i \in \mathbb{R}^{m_i}, i = 1, ..., d, x \in \mathbb{R}^m$.

This smoothness assumption is a classic assumption for RCD methods (Nutini et al., 2015).

A. 2. *(Restricted Strong-Convexity.) With respect to the weighted norm* $\|x\|_L = \sqrt{\sum_{i=1}^d L_i \|x_i\|_2^2}$*, the function* $f(.)$ *and* $g(.)$ *satisfies the following inequality with lower curvature parameter* γ *and tolerance parameter* τ*:*

$$f(x) - f(x^\star) - \langle \nabla f(x^\star), x - x^\star \rangle \geq \frac{\gamma}{2}\|x - x^\star\|_L^2 - \tau g^2(x - x^\star), \tag{7}$$

This form of restricted strong-convexity[1] proposed by Agarwal et al. (2012) encodes the structure-promoting effect of the regularization into the strong-convexity assumption. To be specific, if $\tau = 0$, **A.2** will reduce to the classic strong-convexity assumption normalized with respect to the L-norm (Lin et al., 2015, Assumption 2).

Next we present the basic assumptions of RSC (Agarwal et al., 2012) on the regularization term:

A. 3. *(Subspace Decomposability.) Given a orthogonal subspace pair* $(\mathcal{M}, \mathcal{M}^\perp)$ *in* $\mathbb{R}^m$*,* $g(.)$ *is decomposable if:*

$$g(a + b) = g(a) + g(b), \forall a \in \mathcal{M}, b \in \mathcal{M}^\perp. \tag{8}$$

The subspace $\mathcal{M}$ and $\mathcal{M}^\perp$ are named the *model subspace* and *perturbation subspace* respectively. This property plays a central role in the analytical framework of RSC, and holds true for many structure-promoting regularizers such as the ℓ_1 norm, $\ell_{1,2}$ norm and nuclear norm regularization (Negahban et al., 2012). A related notion of decomposability presented by Vaiter et al. (2015) can extend this work to general gauge functions which will include the analysis priors such as the total-variation regularization.

A. 4. *(Sufficiency of Regularization.) The regularization parameter* λ *satisfies the following inequality with some constant* $c \geq 1$*:*

$$\lambda \geq (1 + \frac{1}{c}) g^\star(\nabla f(x^\dagger)). \tag{9}$$

The sufficient amount of regularization is also a very important requirement in the RSC framework. The intuition of this assumption is simple: in order to have a structured solution $x^\star$, the regularization needs to be strong enough and cannot be arbitrary small. Moreover, Negahban et al. (2012) have shown that with the choice $c = 1$, $\lambda = 2g^\star(\nabla f(x^\dagger))$, the statistical error can be indeed upper bounded and enjoys an optimal scaling w.r.t the sample size and dimension (Negahban et al., 2012, Corollary 2).

For the analysis of Option 2, we need a further assumption namely the "Non-blowout" property in the literature (Lin and Xiao, 2014; Fercoq and Qu, 2017; Wen et al., 2017):

A. 5. *(Non-blowout Iterations.) If we start the APCG algorithm at a point* x_0*, and we assume that there exist a positive constant* $1 \leq \omega < \infty$*, such that the update sequence* $\{x_k\}$ *generated by the algorithm obeys the following inequality almost surely:*

$$F(x_k) - F^\star \leq \omega \left(F(x_0) - F^\star\right), \quad \forall k \tag{10}$$

We assume a relaxed non-blowout property of the APCG iterates, which essentially means that the iterates generated by the algorithm will have optimality gap bounded by that for the first iteration. This assumption hold true for accelerated full gradient and also non-accelerated coordinate descent with $\omega = 1$ which means the iterates are strictly non-blowout. However for accelerated coordinate descent such a result has not been shown and hence we provide it here as a relaxed assumption. Note that the analysis of our Option 1 does not need this assumption.

2.2 Preliminaries for the Analysis

The following definition is useful in our analysis:

Definition 2.2. *(Subspace compatibility.) (Agarwal et al., 2012) With predefined* $g(x)$*, we define the subspace compatibility of a model subspace* $\mathcal{M}$ *as:*

$$\Phi(\mathcal{M}) := \sup_{v \in \mathcal{M} \setminus \{0\}} \frac{g(v)}{\|v\|_L}, \tag{11}$$

when $\mathcal{M} \neq \{0\}$ *and* $\Phi(\{0\}) := 0$.

The subspace compatibility leverages the low-dimensional structure of $x^\star$ into our analysis, for example, if $g(.) = \|.\|_1$, $m = d$, $\|x^\star\|_0 = s$, $L_i = \bar{L}\ \forall i$ and $\mathcal{M}$ is an s-dimensional subspace in $\mathbb{R}^m$, then we have $\Phi(\mathcal{M}) = \sqrt{s/\bar{L}}$.

With the notion of subspace compatibility we are able to provide the key lemma of "effective RSC", which enables us to link the solution's structure with the convergence behavior and quantify their dependence (we provide the proof of this lemma in the supplemental material):

Lemma 2.3. *(Effective RSC) Under* ***A.1 - 4****, if further* ***A.2*** *holds with parameters* (γ, τ) *such that* $\tau\Phi^2(\mathcal{M}) < \frac{\gamma}{16(1+c)^2}$*, then with given* $(x^\star, x^\dagger)$ *and a value* $\eta > 0$*, and denote* $\varepsilon := 2\Phi(\mathcal{M})\|x^\dagger - x^\star\|_2 +$

[1] The RSC was originally defined in the ℓ_2 norm in (Agarwal et al., 2012). We slightly generalize it here by using a weighted norm for the sharper analysis of the coordinate descent algorithms.

$4g(x^{\dagger}_{\mathcal{M}^{\perp}})$ *, for any* x *satisfies* $F(x) - F(x^{\star}) \leq \eta$ *for any optima* $x^{\star}$*, we have:*

$$F(x) - F^{\star} \geq \mu_c \|x - x^{\star}\|_L^2 - 2\tau(1+c)^2 v^2, \qquad (12)$$

where $\mu_c = \frac{\gamma}{2} - 8\tau(1+c)^2\Phi^2(\mathcal{M}) > 0$ *and* $v = \frac{\eta}{\lambda} + \varepsilon$.

We also list the convergence result of the APCG$_0$ which has been proven by Lin et al. (2015):

Lemma 2.4. *(Lin et al., 2015, Theorem 2.1) Under* ***A.1****, the* K_0*-th iteration of APCG*$_0$ *algorithm obeys:*

$$\mathbb{E}F(x_{K_0}) - F^{\star} \leq \left(\frac{2d}{2d+K_0}\right)^2 \mathcal{D}(x^0, x^{\star}) := \Omega_{K_0}, \qquad (13)$$

where $\mathcal{D}(x^0, x^{\star}) := F(x^0) - F^{\star} + \frac{1}{2}\|x^0 - x^{\star}\|_L^2$.

2.3 Main Results

Now we are ready to present our main theorems for our Algorithm 1 with Option 1 and Option 2 in this section, based on the RSC framework.

2.3.1 Convergence Results of Option 1.

We start by our theorem on the objective gap convergence speed of Option 1 which is based on periodic restart scheme:

Theorem 2.5. *Under* ***A.1*** *–* ***4****, if further* ***A.2*** *holds with parameters* (γ, τ) *such that* $\tau\Phi^2(\mathcal{M}) < \frac{\gamma}{16(1+c)^2}$*, and we run the two-stage APCG algorithm (Option 1) with* $K_0 \geq \left\lceil d\left(1 + \frac{2}{\rho\lambda}\right)\sqrt{\frac{8\tau(1+c)^2\mathcal{D}(x^0,x^{\star})}{2\mu_c+1}}\right\rceil$, $K = \left\lceil 2d\beta\sqrt{2 + \frac{1}{\mu_c}} - 2d\right\rceil$ *with* $\beta \geq 2$*, then the following inequality holds:*

$$\mathbb{E}[F(x^{t+1}) - F^{\star}] \leq \max\left\{\varepsilon, \left(\frac{1}{\beta^2}\right)^t \Omega_{K_0}\right\} \qquad (14)$$

with probability at least $1 - \rho$.

We can now summarize the iteration complexity of Option 1 as the following:

Corollary 2.6. *Under the same assumptions and parameter choices of Theorem 2.5, the total number of coordinate gradient calculation of the Two-Stage APCG (Option 1) algorithm needs in order to achieve a* $\delta > \varepsilon$ *objective gap accuracy is:*

$$O\left(\frac{d}{\sqrt{\mu_c}}\right)\log\frac{1}{\delta} + K_0. \qquad (15)$$

We can make the following observations.

(Accelerated Linear Convergence under RSC Framework.) The technical result presented in Theorem 2.5 and Corollary 2.6 demonstrates accelerated linear convergence rate for our two-stage APCG algorithm with Option 1 up to a statistical accuracy under the RSC assumption from Agarwal et al. (2012).

(Structure-Adaptive Convergence.) The effective RSC $\mu_c = \frac{\gamma}{2} - 8\tau(1+c)^2\Phi^2(\mathcal{M})$ provides us a way to link the convergence speed of an algorithm with the structure of the solution. For example, if $c = 1$, $m = d$, $L_i = 1 \ \ \forall i$, $g(x) = \|x\|_1$ and $\|x^{\star}\|_0 = s$, then $\Phi^2(\mathcal{M}) = s$ and hence $\mu_c = \frac{\gamma}{2} - 32\tau s$. Further if $F(x)$ is a Lasso problem, then for a wide class of random design matrix we have $\tau = O(\frac{\log d}{n})$ and $\gamma > 0$. Moreover, Raskutti et al. (2010) have shown that if the data matrix is a correlated Gaussian design matrix such that each row of it is i.i.d drawn from distribution $\mathcal{N}(0, H)$ where H is the covariance matrix and we denote its largest and smallest singular value as $r_{\max}(H)$ and $r_{\min}(H)$, then it can be shown that $\gamma \geq \frac{r_{\min}(H)}{16}$ and $\tau \leq r_{\max}(H)\frac{81\log d}{n}$ with high probability.

(The Early Iterations and High Probability Statement.) From Theorem 2.5 we can see that the probability statement of the convergence result hangs on the choice of the number of iterations on the first stage. Such dependence is natural and within our expectation – the Effective RSC condition presented in Lemma 2.3 is non-vacuous only at a neighborhood of the solution, where the first-stage of our algorithm is aimed to reach.

(Convergence on the Optimization Variable.) Due to the RSC condition we can bound the solution distance to the global optimum by the objective optimality gap (aka, the convergence on the optimization variable). Such results demonstrate that the optimization error on the optimization variable also decays linearly up to a statistical accuracy scaled by a well-behaved constant factor as discussed by Agarwal et al. (2012); Negahban et al. (2012):

Corollary 2.7. *(Convergence of the Iterates) Under the same assumption and parameter choice of Theorem 2.5, the iterates generated by Two-Stage APCG (Option 1) obey the following inequality:*

$$\mathbb{E}\|x^{t+1} - x^{\star}\|_L^2 \leq \left(\frac{1}{\beta^2}\right)^t \frac{\Omega_{K_0}}{\mu_c} + \frac{2\tau(1+c)^2}{\mu_c}\varepsilon^2 + \left(\frac{1}{\beta^4}\right)^t \frac{2\tau(1+c)^2\Omega_{K_0}^2}{\lambda^2\mu_c^2}. \qquad (16)$$

(Connection with Structure-Adaptive Convergence Result for Finite-Sum Optimization.) It is worth noting that this extends the spirit of the recent work Rest-Katyusha (Tang et al., 2018) which is also inspired by and developed under the same RSC framework. The Rest-Katyusha algorithm is a restarted version of an accelerated variance-reduced SGD method

of Allen-Zhu (2017) for efficiently solving regularized empirical risk minimization with a finite-sum structure where $f(x) := \sum_i f_i(x)$ with a smoothness assumption on each f_i, while our coordinate descent method is dedicated to minimizing block-coordinate-wise separable functions with a smoothness assumption on the blocks of coordinates (i.e. **A.1**). Because of this fundamental distinction, we provide here a different complexity result with the RSC framework which complements the contribution provided by Tang et al. (2018).

(The Optimal Choice of β.) For Option 1 of our Two-Stage APCG there is a user defined parameter β. In theory, any $\beta \geq 2$ will provide us an accelerated linear rate. To be specific, to achieve an δ-accuracy, the second stage algorithm needs to have: $\left\lceil 2d\beta\sqrt{2+1/\mu_c} - 2d \right\rceil \log_{\beta^2} \frac{1}{\delta}$ coordinate gradient oracle calls, and hence there is a clear trade-off on β. Similar to (Tang et al., 2018), with some standard calculation one can conclude that the best choice of β to achieve the optimal iteration complexity is roughly the Euler's number (≈ 2.71). We use this choice for our algorithm in the numerical experiments.

2.3.2 Convergence Results of Option 2.

With the additional non-blowout assumption **A.5**, we are also able to provide a similar result for our second approach (with Option 2, we provide the proof of this theorem in the supplemental material):

Theorem 2.8. *Under **A.1** – **5**, and if further **A.2** holds with parameters (γ, τ) such that $\tau\Phi^2(\mathcal{M}) < \frac{\gamma}{16(1+c)^2}$ and we run the Option 2 of the two-stage APCG algorithm with $K = \left\lceil \frac{\log 16}{\log \frac{1}{1-\sqrt{\mu_c}/d}} \right\rceil$ and $K_0 = \left\lceil 8d(1+\frac{\omega}{\lambda\rho})\sqrt{(\sqrt{\frac{1}{\mu_c}}+1)\tau(1+c)^2\mathcal{D}(x^0,x^\star)} \right\rceil$, then the following inequality holds:*

$$\mathbb{E}[F(x^{t+1}) - F^\star] \leq \max\left\{\varepsilon, \left(\frac{1}{4}\right)^t \Omega_{K_0}\right\} \tag{17}$$

with probability at least $1-\rho$.

Again, based on the convergence result on the objective we can summarize the iteration complexity of the Two-Stage APCG algorithm with Option 2 as the following corollary:

Corollary 2.9. *Under the same assumptions and parameter choices of Theorem 2.8, the total number of coordinate gradient calculations the Two-Stage APCG (Option 2) algorithm needs in order to achieve a $\delta > \varepsilon$ objective gap accuracy is:*

$$O\left(\frac{1}{\log \frac{1}{1-\frac{\sqrt{\mu_c}}{d}}}\right)\log\frac{1}{\delta} + K_0. \tag{18}$$

The contraction factor $1 - \frac{\sqrt{\mu_c}}{d}$ occurs in (18) in a logarithmic term $\frac{1}{\log \frac{1}{1-\sqrt{\mu_c}/d}}$ which scales nearly as $\frac{1}{1-(1-\sqrt{\mu_c}/d)} = \frac{d}{\sqrt{\mu_c}}$. Hence we conclude that under the assumptions above, the Two-Stage APCG (Option 2) has a local accelerated linear convergence $O(\frac{d}{\sqrt{\mu_c}}\log\frac{1}{\delta})$.

Because of the RSC condition, the convergence of the iterates can be again easily derived for Option 2 similar to Corollary 2.7 and we do not illustrate this here.

3 ADAPTIVE TWO-STAGE APCG

To the best of our knowledge, all the state-of-the-art accelerated randomized algorithms for solving the composite minimization task (1) require the explicit knowledge of the strong convexity parameter to run with an Nesterov-type accelerated linear convergence rate exactly. For the case where the data fidelity term $f(.)$ is strongly convex, it is difficult in general to calculate the strong convexity parameter before running the accelerated algorithms, let alone in our case, the restricted strong convexity. Here we propose an adaptive restart scheme for Two-Stage APCG based on a heuristic procedure for estimating μ_c on the fly with a small fraction of computational overhead. Similar ideas of adaptive restart have been applied in (O'Donoghue and Candes, 2015; Roulet and d'Aspremont, 2017; Fercoq and Qu, 2017; Tang et al., 2018) for deterministic and stochastic gradient algorithms with Nesterov's acceleration.

(Adaptive Variant of Option 1.). First we observe that for $K = \left\lceil 2d\beta\sqrt{2+1/\mu_c} - 2d \right\rceil$, the convergence speed of the second stage algorithm reads:

$$\mathbb{E}_{\xi_t \setminus \xi_{t-1}} F(x^{t+1}) - F^\star \leq \frac{1}{\beta^2}[F(x^t) - F^\star]. \tag{19}$$

It has been shown by Fercoq and Qu (2017, Prop. 4) that $F(x) - F^\star$ can be lower bounded as $O(\|\mathcal{G}(x) - x\|_2^2)$, where $\mathcal{G}(x)$ is the composite gradient map:

$$\begin{aligned}\mathcal{G}(x) = \arg\min_{u\in\mathbb{R}^d} \frac{d\max_i L_i}{2}\|x-u\|_2^2 \\ + \langle \nabla f(x), u - x\rangle + \lambda g(u).\end{aligned} \tag{20}$$

Meanwhile we can upper bound this objective gap by $O(\|\mathcal{G}(x) - x\|_2^2)$ under some mild assumptions (Fercoq and Qu, 2017). Inspired by such a property, we would like to exploit it as a tool to track the convergence speed of the objective gap, in order to evaluate the accuracy of the RSC parameter of the current iteration. If $\|\mathcal{G}(x^{t+1}) - x^{t+1}\|_2^2 \leq \frac{1}{\beta^2}\|\mathcal{G}(x^t) - x^t\|_2^2$ at t-th iteration, it is likely that we have underestimated the RSC parameter since if $\mu_0 \leq \mu_c$, (19) will always be satisfied. Hence we double the estimate. If otherwise,

Algorithm 2 Adaptive Two-Stage APCG

Inputs: $(x^0, \mu_0, K_0, \beta, T)$

Initialize: $K = \left\lceil 2d\beta\sqrt{2+\frac{1}{\mu_0}} - 2d \right\rceil$;

$x^1 = \text{APCG}_0\ (x^0, K_0)$

Calculate the composite gradient map $\mathcal{G}(x^1)$ by eq:(64).

for $t = 1, \ldots, T$ **do**

$x^{t+1} = \text{APCG}_0\ (x^t, K)$

——Track the convergence speed :

Calculate $\mathcal{G}(x^{t+1})$ by eq:(64)

—— Update the estimate of RSC

if $\|\mathcal{G}(x^{t+1}) - x^{t+1}\|_2^2 \leq \frac{1}{\beta^2}\|\mathcal{G}(x^t) - x^t\|_2^2$

then $\mu_0 \leftarrow 2\mu_0$, **else** $\mu_0 \leftarrow \mu_0/2$.

——Adaptively tune the restart period :

$K = \left\lceil 2d\beta\sqrt{2+\frac{1}{\mu_0}} - 2d \right\rceil$

end for

it is likely that the RSC parameter is overestimated and then we shrink the estimate.

In order to implement the tracking of the objective gap, an extra full gradient is needed to be calculated which will introduce a computational overhead compared to Algorithm 1. However such overhead is durable since the number of restart period K is lower-bounded[2] by $6d$, while the cost of a full gradient is at most d times that of one coordinate gradient calculation, hence the overhead amounts $\frac{1}{6}$ of total iteration complexity at worst.

(Adaptive Variant of Option 2.) The Option 2 of the Two-Stage APCG algorithm can also be made adaptive with a similar idea of utilizing the composite gradient map to estimate the μ_c on the fly. Due to the space limit we include the details of the adaptive variant of Option 2 in the supplemental material.

4 NUMERICAL EXPERIMENTS

This section provides the details of numerical results of our proposed algorithms on solving the Lasso regression problem (Tibshirani, 1996; Tibshirani et al., 2015):

$$x^\star \in \arg\min_{x\in\mathbb{R}^m} \left\{ F(x) := \frac{1}{2n}\|Ax-b\|_2^2 + \lambda\|x\|_1 \right\}, \tag{21}$$

We set all our examples with $A \in \mathbb{R}^{n\times m}$ where $n < m$, hence there is no explicit strong-convexity. We compare our algorithms with state of the art variance-reduced stochastic gradient algorithm Katyusha (Allen-Zhu, 2017, Algorithm 2) which has an accelerated sub-linear convergence rate for non-strongly convex functions, and also the vanilla APCG method for non-strongly-convex functions (Lin et al., 2015, Algorithm 3) as a comparison. We also include the recent Rest-Katyusha algorithm (Tang et al., 2018, Algorithm 1) which also has provable structure-adaptive convergence. For the Rest-Katyusha algorithm and the two choices of our Algorithm 1 which need the explicit knowledge of the RSC parameter, we grid search to estimate it for the best practical performance. We use the theoretical step sizes for our algorithms as well as the APCG in all experiments. For the large datasets (RCV1 and News20) we use minibatch/block-coordinate versions, which are more relevant in parallel-computing scenarios. For the Katyusha and Rest-Katyusha we use the same minibatch size and grid-search the best possible step-sizes to provide the best performance.

Table 1: Chosen Datasets for Lasso Regression

Data set	Size $(n,\ m)$	Reference
Madelon+	(2000, 4000)	Lichman (2013)
Marti2	(500, 1024)	W-team (2008)
RCV1	(20242, 47236)	Lichman (2013)
News20	(15935, 62061)	Rennie (2001)

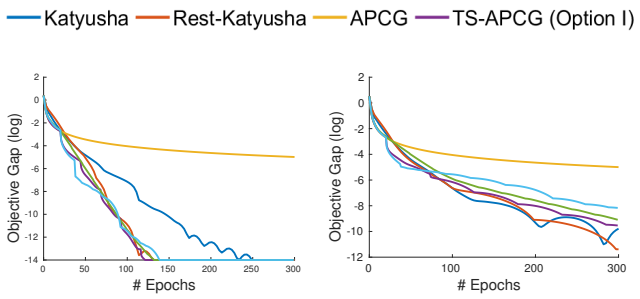
Table 2: Parameter Setting for Alg. 1 and Alg. 2

Experiment	K_0/d	minibatch	μ_0 for Alg.2
Madelon+	20	1	0.1
Marti2	20	1	0.1
RCV1	20	80	0.1
News20	20	100	0.1

For the Madelon dataset we add 3500 random features to its original 500 features. This represents the scenario where one may wish to use sparse regression via an l_1 penalty to nullify the effect of irrelevant features (Langford et al., 2009). For all the four chosen datasets, the Two-Stage APCG algorithm and the adaptive-restart variant significantly outperform the original non-strongly-convex APCG in Lasso regression tasks, and often have superior performance over the Katyusha algorithm. From the results we see that while the original APCG method initially exhibits good objective reduction it has very slow final convergence – this demonstrates the necessity of our two-stage algorithmic structure for the accelerated coordinate descent.

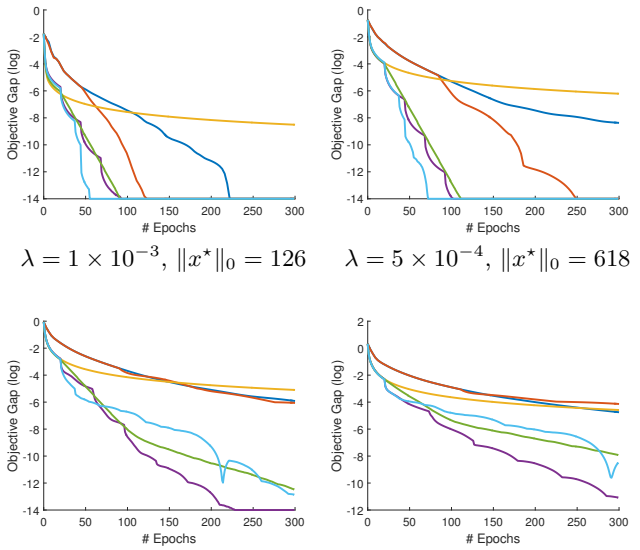
Unlike experiments on the other datasets, for RCV1 dataset, the Katyusha and Rest-Katyusha appear competitive with two-stage APCG. This raises a practical question – for a given dataset, how to choose between the families of primal RCD and SGD (e.g. columns vs. rows). Csiba and Richtárik (2016) provide an analysis comparing the primal RCD and the dual RCD (which also extends to the SGD-type methods in the primal,

[2] by (A.1) and the definition of $\|.\|_L$, we have $\gamma \leq 1$.



$\lambda = 1 \times 10^{-4}$, $\|x^\star\|_0 = 902$ $\lambda = 5 \times 10^{-5}$, $\|x^\star\|_0 = 1653$

Figure 1: Lasso Regression on RCV1 Dataset



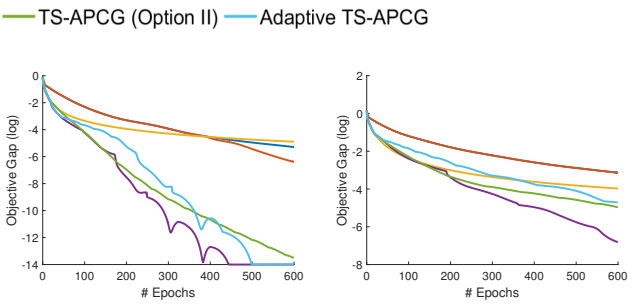
$\lambda = 1 \times 10^{-3}$, $\|x^\star\|_0 = 126$ $\lambda = 5 \times 10^{-4}$, $\|x^\star\|_0 = 618$

$\lambda = 2 \times 10^{-4}$, $\|x^\star\|_0 = 1250$ $\lambda = 1 \times 10^{-4}$, $\|x^\star\|_0 = 1594$

Figure 2: Lasso Regression on Madelon Dataset with 3500 Additional Random Features

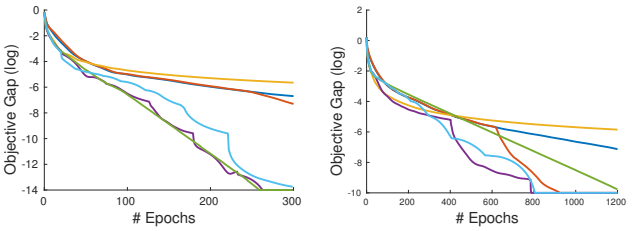
see Shalev-Shwartz (2016)). Although restricted to ℓ_2 regularization, their analysis suggests that the complexity of primal RCD and dual RCD is dependant on the dataset's characteristics such as the density and the distribution of the features. Using their complexity bounds we found that in theory the RCV1 and News20 dataset prefer dual RCD for ℓ_2 regularized ERM, while the Madelon and Marti2 prefer primal RCD, which is in broad agreement with our Lasso results.

These numerical results on real data sets have demonstrated the effectiveness of our approaches for accelerating the APCG method via actively exploiting the low dimensional structure of the solution. Non-structure-adaptive accelerated methods like Katyusha and APCG are blind the restricted strong convexity. Hence when the solution is relatively sparse, or rather, the regularization parameter is relatively large for the data set, the two-stage APCG algorithms enjoy local linear



$\lambda = 2 \times 10^{-5}$, $\|x^\star\|_0 = 48$ $\lambda = 5 \times 10^{-6}$, $\|x^\star\|_0 = 119$

Figure 3: Lasso Regression on MARTI2 Dataset



$\lambda = 5 \times 10^{-5}$, $\|x^\star\|_0 = 267$ $\lambda = 1 \times 10^{-5}$, $\|x^\star\|_0 = 1837$

Figure 4: Lasso Regression on the News20 Dataset (Class 1).

convergence and often significantly outperform these baselines. Moreover the our adaptive two-stage APCG algorithm appears to be very successful in estimating the RSC parameter and adaptively tuning the restart period on the fly such that it achieves comparable convergence speed to the two-stage APCG methods which need a reliable RSC estimate beforehand.

5 CONCLUSION

In this work, we provide theoretical and algorithmic contributions to coordinate descent optimization. We analyze the structure-adaptive convergence of a simple variant (namely the Two-stage APCG) of accelerated RCD based on the RSC framework of Agarwal et al. (2012). Moreover, we propose an adaptive-restart that does not require the explicit knowledge of RSC but estimates it on the fly. We validate the effectiveness of our approach via numerical experiments on sparse regression tasks. This work opens up the potential to develop even faster structure-adaptive accelerated coordinate descent methods incorporating importance sampling (Allen-Zhu et al., 2016) for better iteration complexity, screening-rules (Ndiaye et al., 2016) to predict the zero-elements for sparse regression and skip redundant updates, and continuation methods (Lin and Xiao, 2014) for even faster initial convergence, etc.

6 Acknowledgements

JT, FB and MD would like to acknowledge the support from H2020-MSCA-ITN Machine Sensing Training Network (MacSeNet), project 642685; ERC grant SEQUOIA; and ERC Advanced grant, project 694888, C-SENSE, respectively. MD is also supported by a Royal Society Wolfson Research Merit Award. JT would like to thank Damien Scieur and Vincent Roulet for helpful discussions during his research visit in SIERRA team.

References

Agarwal, A., Negahban, S. and Wainwright, M. J. (2012), 'Fast global convergence rates of gradient methods for high-dimensional statistical recovery', *The Annals of Statistics* **40**(5), 2452–2482.

Allen-Zhu, Z. (2017), Katyusha: The first direct acceleration of stochastic gradient methods, *in* 'Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing', ACM, pp. 1200–1205.

Allen-Zhu, Z., Qu, Z., Richtárik, P. and Yuan, Y. (2016), Even faster accelerated coordinate descent using non-uniform sampling, *in* 'International Conference on Machine Learning', pp. 1110–1119.

Arjevani, Y. (2017), Limitations on variance-reduction and acceleration schemes for finite sums optimization, *in* 'Advances in Neural Information Processing Systems', pp. 3543–3552.

Bach, F., Jenatton, R., Mairal, J., Obozinski, G. et al. (2012), 'Optimization with sparsity-inducing penalties', *Foundations and Trends® in Machine Learning* **4**(1), 1–106.

Bickel, P. J., Li, B., Tsybakov, A. B., van de Geer, S. A., Yu, B., Valdés, T., Rivero, C., Fan, J. and van der Vaart, A. (2006), 'Regularization in statistics', *Test* **15**(2), 271–344.

Bühlmann, P. and Van De Geer, S. (2011), *Statistics for high-dimensional data: methods, theory and applications*, Springer Science & Business Media.

Csiba, D. and Richtárik, P. (2016), 'Coordinate descent face-off: Primal or dual?', *arXiv preprint arXiv:1605.08982* .

Fercoq, O. and Qu, Z. (2016), 'Restarting accelerated gradient methods with a rough strong convexity estimate', *arXiv preprint arXiv:1609.07358* .

Fercoq, O. and Qu, Z. (2017), 'Adaptive restart of accelerated gradient methods under local quadratic growth condition', *arXiv preprint arXiv:1709.02300* .

Fercoq, O. and Qu, Z. (2018), 'Restarting the accelerated coordinate descent method with a rough strong convexity estimate', *arXiv preprint arXiv:1803.05771* .

Fercoq, O. and Richtárik, P. (2015), 'Accelerated, parallel, and proximal coordinate descent', *SIAM Journal on Optimization* **25**(4), 1997–2023.

Hsieh, C.-J., Chang, K.-W., Lin, C.-J., Keerthi, S. S. and Sundararajan, S. (2008), A dual coordinate descent method for large-scale linear svm, *in* 'Proceedings of the 25th international conference on Machine learning', ACM, pp. 408–415.

Langford, J., Li, L. and Zhang, T. (2009), 'Sparse online learning via truncated gradient', *Journal of Machine Learning Research* **10**(Mar), 777–801.

Lee, Y. T. and Sidford, A. (2013), Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems, *in* 'Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on', IEEE, pp. 147–156.

Lichman, M. (2013), 'UCI machine learning repository'. **URL:** *http://archive.ics.uci.edu/ml*

Lin, Q., Lu, Z. and Xiao, L. (2014), An accelerated proximal coordinate gradient method, *in* 'Advances in Neural Information Processing Systems', pp. 3059–3067.

Lin, Q., Lu, Z. and Xiao, L. (2015), 'An accelerated randomized proximal coordinate gradient method and its application to regularized empirical risk minimization', *SIAM Journal on Optimization* **25**(4), 2244–2273.

Lin, Q. and Xiao, L. (2014), An adaptive accelerated proximal gradient method and its homotopy continuation for sparse optimization, *in* 'International Conference on Machine Learning', pp. 73–81.

Lu, H., Freund, R. and Mirrokni, V. (2018), Accelerating greedy coordinate descent methods, *in* 'Proceedings of the 35th International Conference on Machine Learning', Vol. 80 of *Proceedings of Machine Learning Research*, PMLR, pp. 3257–3266.

Lu, Z. and Xiao, L. (2015), 'On the complexity analysis of randomized block-coordinate descent methods', *Mathematical Programming* **152**(1-2), 615–642.

Ndiaye, E., Fercoq, O., Gramfort, A. and Salmon, J. (2016), 'Gap safe screening rules for sparsity enforcing penalties', *arXiv preprint arXiv:1611.05780* .

Negahban, S. N., Ravikumar, P., Wainwright, M. J. and Yu, B. (2012), 'A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers', *Statistical Science* pp. 538–557.

Nesterov, Y. (1983), A method of solving a convex programming problem with convergence rate o (1/k2), *in* 'Soviet Mathematics Doklady', Vol. 27, pp. 372–376.

Nesterov, Y. (2007), Gradient methods for minimizing composite objective function, Technical report, UCL.

Nesterov, Y. (2012), 'Efficiency of coordinate descent methods on huge-scale optimization problems', *SIAM Journal on Optimization* **22**(2), 341–362.

Nesterov, Y. (2013), *Introductory lectures on convex optimization: A basic course*, Vol. 87, Springer Science & Business Media.

Nesterov, Y. and Stich, S. U. (2017), 'Efficiency of the accelerated coordinate descent method on structured optimization problems', *SIAM Journal on Optimization* **27**(1), 110–123.

Nutini, J., Schmidt, M., Laradji, I., Friedlander, M. and Koepke, H. (2015), Coordinate descent converges faster with the gauss-southwell rule than random selection, *in* 'International Conference on Machine Learning', pp. 1632–1641.

O'Donoghue, B. and Candes, E. (2015), 'Adaptive restart for accelerated gradient schemes', *Foundations of computational mathematics* **15**(3), 715–732.

Qin, Z., Scheinberg, K. and Goldfarb, D. (2013), 'Efficient block-coordinate descent algorithms for the group lasso', *Mathematical Programming Computation* **5**(2), 143–169.

Qu, C. and Xu, H. (2016), 'Linear convergence of svrg in statistical estimation', *arXiv preprint arXiv:1611.01957* .

Qu, C. and Xu, H. (2017), 'Linear convergence of sdca in statistical estimation', *arXiv preprint arXiv:1701.07808* .

Raskutti, G., Wainwright, M. J. and Yu, B. (2010), 'Restricted eigenvalue properties for correlated gaussian designs', *Journal of Machine Learning Research* **11**(Aug), 2241–2259.

Rennie, J. D. (2001), 'Improving multi-class text classification with naive bayes'.

Richtárik, P. and Takáč, M. (2014), 'Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function', *Mathematical Programming* **144**(1-2), 1–38.

Roulet, V. and d'Aspremont, A. (2017), Sharpness, restart and acceleration, *in* 'Advances in Neural Information Processing Systems', pp. 1119–1129.

Shalev-Shwartz, S. (2016), Sdca without duality, regularization, and individual convexity, *in* 'International Conference on Machine Learning', pp. 747–754.

Tang, J., Bach, F., Golbabaee, M. and Davies, M. (2017), 'Structure-adaptive, variance-reduced, and accelerated stochastic optimization', *arXiv preprint arXiv:1712.03156* .

Tang, J., Golbabaee, M., Bach, F. and Davies, M. (2018), 'Rest-katyusha: Exploiting the solution's structure via scheduled restart schemes', *arXiv preprint arXiv:1803.02246. To appear in NIPS 2018* .

Tibshirani, R. (1996), 'Regression shrinkage and selection via the lasso', *Journal of the Royal Statistical Society. Series B (Methodological)* pp. 267–288.

Tibshirani, R., Wainwright, M. and Hastie, T. (2015), *Statistical learning with sparsity: the lasso and generalizations*, Chapman and Hall/CRC.

Vaiter, S., Golbabaee, M., Fadili, J. and Peyré, G. (2015), 'Model selection with low complexity priors', *Information and Inference: A Journal of the IMA* **4**(3), 230–287.

Vapnik, V. (2013), *The nature of statistical learning theory*, Springer science & business media.

W-team, C. (2008), 'Measurement artifacts'.
URL: *http://www.causality.inf.ethz.ch/data/MARTI.html*

Wainwright, M. J. (2014), 'Structured regularizers for high-dimensional problems: Statistical and computational issues', *Annual Review of Statistics and Its Application* **1**, 233–253.

Wen, B., Chen, X. and Pong, T. K. (2017), 'Linear convergence of proximal gradient algorithm with extrapolation for a class of nonconvex nonsmooth minimization problems', *SIAM Journal on Optimization* **27**(1), 124–145.

Wen, Z., Goldfarb, D. and Scheinberg, K. (2012), Block coordinate descent methods for semidefinite programming, *in* 'Handbook on Semidefinite, Conic and Polynomial Optimization', Springer, pp. 533–564.

Wright, S. J. (2015), 'Coordinate descent algorithms', *Mathematical Programming* **151**(1), 3–34.

Wu, T. T., Lange, K. et al. (2008), 'Coordinate descent algorithms for lasso penalized regression', *The Annals of Applied Statistics* **2**(1), 224–244.

Appendix

A The proof for Option 1

A.1 The Proof for Lemma 2.3

The proof of this lemma follows:

Proof. Let us denote $\Delta = x - x^\dagger$. Since we have assumed $F(x) - F(x^\star) \leq \eta$, then we also have $F(x) - F(x^\dagger) \leq \eta$, hence:

$$f(x^\dagger + \Delta) + \lambda g(x^\dagger + \Delta) \leq f(x^\dagger) + \lambda g(x^\dagger) + \eta, \quad (22)$$

then substract both side with $\langle \nabla f(x^\dagger), \Delta \rangle$ and rearrange:

$$\begin{aligned} & f(x^\dagger + \Delta) - f(x^\dagger) - \langle \nabla f(x^\dagger), \Delta \rangle \\ & + \lambda g(x^\dagger + \Delta) - \lambda g(x^\dagger) \\ & \leq -\langle \nabla f(x^\dagger), \Delta \rangle + \eta. \end{aligned} \quad (23)$$

Due to the convexity of $f(.)$ we immediately have:

$$\begin{aligned} \lambda g(x^\dagger + \Delta) - \lambda g(x^\dagger) &\leq -\langle \nabla f(x^\dagger), \Delta \rangle + \eta \\ &\leq g^*(\nabla f(x^\dagger)) g(\Delta) + \eta \\ &\leq \frac{\lambda}{1 + \frac{1}{c}} g(\Delta) + \eta, \end{aligned}$$

hence by dividing both side with λ and then applying the decomposability of g we have:

$$g(x^\dagger + \Delta) - g(x^\dagger) \leq \frac{1}{1 + \frac{1}{c}} [g(\Delta_{\mathcal{M}}) + g(\Delta_{\mathcal{M}^\perp})] + \frac{\eta}{\lambda}, \quad (24)$$

and meanwhile the lower bound on the left-hand-side has been provided in (Agarwal et al., 2012), which reads:

$$g(x^\dagger + \Delta) - g(x^\dagger) \geq g(\Delta_{\mathcal{M}^\perp}) - 2g(x^\dagger_{\mathcal{M}^\perp}) - g(\Delta_{\mathcal{M}}). \quad (25)$$

By combining these two bounds we have:

$$\begin{aligned} & g(\Delta_{\mathcal{M}^\perp}) + g(\Delta_{\mathcal{M}}) + \frac{(1 + \frac{1}{c})\eta}{\lambda} \\ & \leq (1 + \frac{1}{c}) g(\Delta_{\mathcal{M}^\perp}) - 2(1 + \frac{1}{c}) g(x^\dagger_{\mathcal{M}^\perp}) \\ & - (1 + \frac{1}{c}) g(\Delta_{\mathcal{M}}), \end{aligned} \quad (26)$$

and then:

$$\begin{aligned} \frac{1}{c} g(\Delta_{\mathcal{M}^\perp}) &\leq (2 + \frac{1}{c}) g(\Delta_{\mathcal{M}}) + 2(1 + \frac{1}{c}) g(x^\dagger_{\mathcal{M}^\perp}) \\ &\quad + \frac{(1 + \frac{1}{c})\eta}{\lambda} \\ g(\Delta_{\mathcal{M}^\perp}) &\leq (1 + 2c) g(\Delta_{\mathcal{M}}) + 2(1 + c) g(x^\dagger_{\mathcal{M}^\perp}) \\ &\quad + \frac{(1 + c)\eta}{\lambda} \\ g(\Delta) &\leq (2 + 2c)(g(\Delta_{\mathcal{M}}) + g(x^\dagger_{\mathcal{M}^\perp})) + \frac{(1 + c)\eta}{\lambda} \end{aligned}$$

Now let $\Delta_x := x - x^\star$ where x satisfies $F(x) - F(x^\star) \leq \eta$, and $\Delta^\star := x^\star - x^\dagger$. Due to the fact that $x^\star$ is the optimal point, η can be set as 0 if $x = x^\star$, then:

$$g(\Delta^\star) \leq (2 + 2c)(g(\Delta^\star_{\mathcal{M}}) + g(x^\dagger_{\mathcal{M}^\perp})), \quad (27)$$

and now we are able to bound $g(\Delta_x)$:

$$\begin{aligned} & g(\Delta_x) \\ \leq\ & g(\Delta) + g(\Delta^\star) \\ \leq\ & (2 + 2c) g(\Delta_{\mathcal{M}}) + (2 + 2c) g(\Delta^\star_{\mathcal{M}}) \\ & + (4 + 4c) g(x^\dagger_{\mathcal{M}^\perp}) + \frac{(1 + c)\eta}{\lambda} \\ \leq\ & (1 + c) \left[2g(\Delta_{\mathcal{M}}) + 2g(\Delta^\star_{\mathcal{M}}) + 4g(x^\dagger_{\mathcal{M}^\perp}) + \frac{\eta}{\lambda} \right]. \end{aligned}$$

then by the definition of the subspace compatibility $\Phi(\mathcal{M}) := \sup_{v \in \mathcal{M} \setminus \{0\}} \frac{g(v)}{\|v\|_L}$ we can write:

$$\begin{aligned} & g(\Delta_x) \\ =\ & g(x - x^\star) \\ \leq\ & (1 + c)[2\Phi(\mathcal{M}) \|x - x^\star\|_L + 2\Phi(\mathcal{M}) \|x^\dagger - x^\star\|_L \\ & + 4g(x^\dagger_{\mathcal{M}^\perp}) + \frac{\eta}{\lambda}] \\ \leq\ & (1 + c) \left[2\Phi(\mathcal{M}) \|x - x^\star\|_L + v \right], \end{aligned}$$

where we denote $\varepsilon := 2\Phi(\mathcal{M}) \|x^\dagger - x^\star\|_L + 4g(x^\dagger_{\mathcal{M}^\perp})$ and $v := \frac{\eta}{\lambda} + \varepsilon$. Then because of the fact that $(a+b)^2 \leq 2a^2 + 2b^2$ we have:

$$g^2(x - x^\star) \leq (1+c)^2 \left[8\Phi^2(\mathcal{M}) \|x - x^\star\|_L^2 + 2v^2 \right]. \quad (28)$$

Due to **A.2** we can write:

$$\begin{aligned} & f(x) - f(x^\star) - \langle \nabla f(x^\star), x - x^\star \rangle \\ \geq\ & \frac{\gamma}{2} \|x - x^\star\|_L^2 + \tau(1 + c)^2 \left[8\Phi^2(\mathcal{M}) \|x - x^\star\|_L^2 + 2v^2 \right] \\ \geq\ & \left[\frac{\gamma}{2} - 8\tau(1 + c)^2 \Phi^2(\mathcal{M}) \right] \|x - x^\star\|_L^2 - 2\tau(1 + c)^2 v^2, \end{aligned}$$

Then because $g(.)$ is convex, we can write:

$$g(x) - g(x^\star) - \langle \partial g(x^\star), x - x^\star \rangle \geq 0, \quad (29)$$

and we have:

$$\begin{aligned} & F(x) - F^\star - \langle \nabla f(x^\star) + \partial g(x^\star), x - x^\star \rangle \\ \geq & \left[\frac{\gamma}{2} - 8\tau(1+c)^2 \Phi^2(\mathcal{M}) \right] \|x - x^\star\|_L^2 - 2\tau(1+c)^2 v^2. \end{aligned}$$

By first order optimality condition we have $\langle \nabla f(x^\star) + \partial g(x^\star), x - x^\star \rangle \geq 0$, hence we justify the claim. □

A.2 The Proof for Theorem 2.5, Corollary 2.6 and 2.7

Proof. We first define a sequence of random variable ξ_t which is the realization of the random choices of coordinates from the 0-th iteration to the end of t-th iteration of Two-stage APCG (Option 1). According to Lemma 2, after the first stage we have:

$$\mathbb{E}_{\xi_0} F(x^1) - F^\star \leq \epsilon_1 := \Omega_{K_0}. \tag{30}$$

Then with Markov inequality, at a probability at least $1 - \frac{\rho}{2}$ we have:

$$F(x^1) - F^\star \leq \frac{2}{\rho} \epsilon_1. \tag{31}$$

Now we define three shrinking sequence through which we will achieve the proof via induction: $\epsilon_{t+1} = \frac{1}{\beta^2} \epsilon_t$, $\rho_{t+1} = \frac{1}{\beta} \rho_t$ with $\rho_1 := \rho$, and $v_t = \frac{2\epsilon_t}{\lambda \rho_t} + \varepsilon$.

Induction step 1: we first reformulate the effective RSC presented in Lemma 1 as the following:

$$\|x - x^\star\|_L^2 \leq \frac{1}{\mu_c} \left[F(x) - F^\star + 2\tau(1+c)^2 v^2 \right], \tag{32}$$

then we can have:

$$\begin{aligned} & \mathbb{E}_{\xi_1 \setminus \xi_0} F(x^2) - F^\star \\ \leq & \left(\frac{2d}{2d+K} \right)^2 [F(x^1) - F^\star] \\ & + \left(\frac{2d}{2d+K} \right)^2 \frac{1}{2\mu_c} [F(x^1) - F^\star + 2\tau(1+c)^2 v_1^2] \\ = & \frac{4d^2 + \frac{2d^2}{\mu_c}}{(k+2d)^2} [F(x^1) - F^\star] + \frac{4d^2 \tau (1+c)^2 v_1^2}{\mu_c (2d+K)^2}. \end{aligned}$$

By taking expectation on both sides over ξ_0, we have:

$$\begin{aligned} & \mathbb{E}_{\xi_1} F(x^2) - F^\star \\ \leq & \frac{4d^2 + \frac{2d^2}{\mu_c}}{(2d+K)^2} \epsilon_1 + \frac{4d^2 \tau (1+c)^2 v_1^2}{\mu_c (2d+K)^2} \\ \leq & \frac{4d^2 + \frac{2d^2}{\mu_c}}{(2d+K)^2} \epsilon_1 + \frac{4d^2 \tau (1+c)^2}{\mu_c (2d+K)^2} \left(\frac{2\epsilon_1}{\rho \lambda} + \epsilon_1 \right)^2, \end{aligned}$$

where the second inequality holds due to $\epsilon_t > \varepsilon \ \ \forall t$. Then we set:

$$\frac{4d^2 \tau (1+c)}{\mu_c} \left(\frac{2\epsilon_1}{\rho \lambda} + \epsilon_1 \right)^2 \leq (4d^2 + \frac{2d^2}{\mu_c}) \epsilon_1, \tag{33}$$

hence:

$$\left(\frac{2}{\rho \lambda} + 1 \right)^2 \epsilon_1 \leq \frac{2\mu_c + 1}{2\tau (1+c)^2}. \tag{34}$$

Since $\epsilon_1 = \frac{4d^2 \mathcal{D}(x^0, x^\star)}{(2d+K_0)^2}$, it is enough to set:

$$K_0 = \left\lceil d \left(\frac{2}{\rho \lambda} + 1 \right) \sqrt{\frac{8\tau(1+c)^2 \mathcal{D}(x^0, x^\star)}{2\mu_c + 1}} \right\rceil, \tag{35}$$

to ensure that:

$$\mathbb{E}_{\xi_1} F(x^2) - F^\star \leq \frac{8d^2 + \frac{4d^2}{\mu_c}}{(K+2d)^2} \epsilon_1. \tag{36}$$

Then if we choose:

$$K = \left\lceil 2d\beta \sqrt{2 + \frac{1}{\mu_c}} - 2d \right\rceil, \tag{37}$$

we can ensure that:

$$\mathbb{E}_{\xi_1} F(x^2) - F^\star \leq \frac{1}{\beta^2} \epsilon_1. \tag{38}$$

Induction step 2: At iteration $t+1$, due to the induction hypothesis $\mathbb{E}_{\xi_{t-1}} F(x^t) - F^\star \leq \epsilon_t = \frac{\epsilon_{t-1}}{\beta^2}$ we have:

$$\begin{aligned} & \mathbb{E}_{\xi_t} F(x^{t+1}) - F^\star \\ \leq & \frac{4d^2 + \frac{2d^2}{\mu_c}}{(2d+K)^2} \mathbb{E}_{\xi_{t-1}} [F(x^t) - F^\star] + \frac{4d^2 \tau (1+c)^2 v_t^2}{\mu_c (2d+K)^2} \\ = & \frac{4d^2 + \frac{2d^2}{\mu_c}}{(2d+K)^2} \epsilon_t + \frac{4d^2 \tau (1+c)^2 v_t^2}{\mu_c (2d+K)^2} \\ \leq & \frac{4d^2 + \frac{2d^2}{\mu_c}}{(2d+K)^2} \epsilon_t + \frac{4d^2 \tau (1+c)^2}{\mu_c (2d+K)^2} \left(\frac{2\epsilon_t}{\rho_t \lambda} + \epsilon_t \right)^2, \end{aligned}$$

Then we set:

$$\frac{4d^2 \tau (1+c)}{\mu_c} \left(\frac{2\epsilon_1}{\rho_t \lambda} + \epsilon_t \right)^2 \leq (4d^2 + \frac{2d^2}{\mu_c}) \epsilon_t, \tag{39}$$

and reformulate it as:

$$\left(\frac{2}{\rho_t \lambda} + 1 \right)^2 \epsilon_t \leq \frac{2\mu_c + 1}{2\tau (1+c)^2}. \tag{40}$$

Since we have chosen $\rho_t = \frac{1}{\beta} \rho_{t-1}$, $\epsilon_t = \frac{1}{\beta^2} \epsilon_{t-1}$ with $\beta \geq 2$,

$$\left(\frac{2}{\rho_t \lambda} + 1 \right)^2 \epsilon_t \leq \left(\frac{2}{\rho_{t-1} \lambda} + 1 \right)^2 \epsilon_{t-1} \leq \left(\frac{2}{\rho \lambda} + 1 \right)^2 \epsilon_1, \tag{41}$$

hence with the same choice of K_0 and K in induction step 1, with probability at least $1 - \sum_{i=1}^t \frac{\rho_i}{2} \geq 1 - \frac{\rho\beta}{2(\beta-1)} \geq 1 - \rho$ (due to the choice $\beta \geq 2$), we can ensure:

$$\mathbb{E}_{\xi_t} F(x^{t+1}) - F^\star \leq \frac{1}{\beta^2} \epsilon_t. \tag{42}$$

Thus finishes the proof of Theorem 2.5.

In summary, to achieve $\mathbb{E}_{\xi_t} F(x^{t+1}) - F^\star \leq \delta$, the coordinate gradient calculation at the second stage should be:

$$\left\lceil 2d\beta\sqrt{2+\frac{1}{\mu_c}} - 2d \right\rceil \log_{\beta^2} \frac{1}{\delta}, \tag{43}$$

and we justifies the claim in Corollary 2.6.

We can also provide the convergence result of the optimization variable by the Effective RSC given by Lemma 2.3. At point x^{T+1}, we set $\eta = F(x^{T+1}) - F^\star$, and we have:

$$\begin{aligned}
& \|x^{T+1} - x^\star\|_L^2 \\
\leq\ & \frac{F(x^{T+1}) - F^\star + 2\tau(1+c)^2 v^2}{\mu_c} \\
\leq\ & \frac{F(x^{T+1}) - F^\star + 2\tau(1+c)^2[(\frac{\eta}{\lambda})^2 + \varepsilon^2]}{\mu_c} \\
\leq\ & \frac{F(x^{T+1}) - F^\star}{\mu_c} + \frac{2\tau(1+c)^2}{\lambda^2\mu_c}\left(F(x^{T+1}) - F^\star\right)^2 \\
& + \frac{2\tau(1+c)^2\varepsilon^2}{\mu_c} \\
\leq\ & \left(\frac{1}{\beta^2}\right)^T \frac{\Omega_{K_0}}{\mu_c} + \frac{2\tau(1+c)^2\Omega_{K_0}^2}{\lambda^2\mu_c^2}\left(\frac{1}{\beta^4}\right)^T \\
& + \frac{2\tau(1+c)^2}{\lambda^2\mu_c}\varepsilon^2.
\end{aligned}$$

Hence til now we finish the proofs for both Theorem 2.5, Corollary 2.6 and Corollary 2.7. □

B Convergence proof for Option 2

First we present a key lemma for two-stage with Option 2, which is extended from the convergence proof of (Lin et al., 2014, 2015):

Lemma B.1. *Given $(x^\star, x^\dagger)$, and denote $\varepsilon := 2\Phi(\mathcal{M})\|x^\dagger - x^\star\|_L + 4g(x^\dagger_{\mathcal{M}^\perp})$, if the regularization parameter λ and the reference point $x^\dagger$ satisfy $\lambda \geq (1+\frac{1}{c})g^*(\nabla f(x^\dagger))$. Assume that the non-blowout assumption holds with parameter ω, the updates of the second stage of the Two-Stage APCG obeys:*

$$\mathbb{E}_{\xi_K^t \backslash \xi_K^{t-1}}[F(x^{t+1})] - F^\star \leq \left(1 - \frac{\sqrt{\mu_c}}{d}\right)^K \cdot 2\left[F(x^t) - F^\star\right] + 2\tau(1+c)^2\left(\sqrt{\frac{1}{\mu_c}} + 1\right)v^2, \tag{44}$$

where $\mu_c = \frac{\gamma}{2} - 8\tau(1+c)^2\Phi^2(\mathcal{M})$, $v = \frac{\eta}{\lambda} + \varepsilon$, $F(x_k^t) - F(x^\star) \leq \eta := \omega\left(F(x_0^t) - F^\star\right)$ for all $t \geq 1$ and k.

Proof. At each iteration, the APCG algorithm chooses a coordinate uniformly at random to perform updates. The update sequences x_{k+1}^t and z_{k+1}^t depend on the realization of the following random variable which we denote as ξ_k^t:

$$\xi_k^t = \{i_k^t, i_{k-1}^t, ..., i_1^t, i_0^t, i_k^{t-1}, ..., i_0^{t-1}, ..., i_k^0, ..., i_0^0\}, \tag{45}$$

and for the randomness within a single outer-loop of Two-Stage APCG we specifically denote $\xi_k^t \backslash \xi_k^{t-1}$ as

$$\xi_k^t \backslash \xi_k^{t-1} = \{i_k^t, i_{k-1}^t, ..., i_1^t, i_0^t\} \tag{46}$$

We achieve the proof of this lemma by extending the original proof for strongly-convex APCG (Lin et al., 2015, Theorem 2.1), that there is only one place the strong-convexity assumption on $f(x)$ is used (after equation 3.20). Hence by replacing the original strong-convexity with the effective RSC we have the following:

$$\begin{aligned}
& \mathbb{E}_{i_k^t}[f(x_{k+1}^t) + \lambda\hat{g}_{k+1}^t - F^\star + \frac{\mu_c}{2}\|z_{k+1}^t - x^\star\|_L^2] \\
\leq\ & \left(1 - \frac{\sqrt{\mu_c}}{d}\right)\mathbb{E}_{i_{k-1}^t}[f(x_k^t) + \lambda\hat{g}_k^t - F^\star + \frac{\mu_c}{2}\|z_k^t - x^\star\|_L^2] \\
& + \frac{2\tau(1+c)^2}{d}v^2,
\end{aligned}$$

(the detailed definition of $\hat{g}_k^t$ can be found in (Lin et al., 2015, Lemma 3.3), which is a convex combination of $g(z_0^t)$, $g(z_1^t)$, $g(z_2^t)$ ..... $g(z_k^t)$) and then we roll up the bound:

$$\begin{aligned}
& \mathbb{E}_{\xi_k^t \backslash \xi_K^{t-1}}[f(x_{k+1}^t) + \lambda\hat{g}_{k+1}^t - F^\star + \frac{\mu_c}{2}\|z_{k+1}^t - x^\star\|_L^2] \\
\leq\ & \left(1 - \frac{\sqrt{\mu_c}}{d}\right)^k [F(x_0^t) - F^\star + \frac{\mu_c}{2}\|x_0^t - x^\star\|_L^2] \\
& + \frac{1 - (1 - \sqrt{\mu_c}/d)^{k-1}}{1 - (1 - \sqrt{\mu_c}/d)} \frac{2\tau(1+c)^2}{d} v^2 \\
\leq\ & \left(1 - \frac{\sqrt{\mu_c}}{d}\right)^k [F(x_0^t) - F^\star + \frac{\mu_c}{2}\|x_0^t - x^\star\|_L^2] \\
& + \frac{2\tau(1+c)^2}{\sqrt{\mu_c}} v^2 \\
\leq\ & \left(1 - \frac{\sqrt{\mu_c}}{d}\right)^k [2F(x_0^t) - 2F^\star + 2(1+c)^2\tau v^2] \\
& + \frac{2\tau(1+c)^2}{\sqrt{\mu_c}} v^2 \\
\leq\ & \left(1 - \frac{\sqrt{\mu_c}}{d}\right)^k \cdot 2\left[F(x_0^t) - F^\star\right] \\
& + 2\tau(1+c)^2\left(\sqrt{\frac{1}{\mu_c}} + 1\right)v^2
\end{aligned}$$

where we utilize the effective RSC again to bound the term $\frac{\mu_c}{2}\|x_0^t - x^\star\|_L^2$.

Since $\hat{g}_{k+1}^t \geq g(x_{k+1}^t)$ as declared in (Lin et al., 2015), by simplifying the left hand side we can have:

$$\mathbb{E}_{\xi_k^t \setminus \xi_K^{t-1}}[F(x_{k+1}^t)] - F^\star \leq \left(1 - \frac{\sqrt{\mu_c}}{d}\right)^K \cdot 2\left[F(x_0^t) - F^\star\right] + 2\tau(1+c)^2\left(\sqrt{\frac{1}{\mu_c}} + 1\right)v^2. \tag{47}$$

Thus finishes the proof since $F(x_0^{t+1}) = F(x_{K+1}^t)$. □

B.1 Proof of Theorem 2.8 and Corollary 2.9

Then we are ready to present the proof of Theorem 2.

Proof. We follow a similar procedure by Agarwal et al. (2012) and Qu and Xu (2016) to roll up the residual term v^2. According to Lin et al. (2015) for the first stage of the algorithm we have:

$$\mathbb{E}_{\xi^0}[F(x^1)] - F^\star \leq \epsilon_1 := \left(\frac{2d}{2d + K_0}\right)^2 \mathcal{D}(x^0, x^\star),$$

where $\mathcal{D}(x^0, x^\star) := [F(x^0) - F^\star + \frac{1}{2}\|x^0 - x^\star\|_L^2$. Then with Markov inequality, at a probability at least $1 - \frac{\rho}{2}$ we have:

$$F(x^1) - F^\star \leq \frac{2}{\rho}\epsilon_1. \tag{48}$$

Next we derive the complexity of the second stage. We define three shrinking sequence through which we will achieve the proof via induction: $\epsilon_{t+1} = \frac{1}{4}\epsilon_t$, $\rho_{t+1} = \frac{1}{2}\rho_t$ with $\rho_1 := \rho$, and $v_t = \frac{\omega\epsilon_t}{\lambda\rho_t} + \varepsilon$.

Induction part 1: We turn to our first outer iteration in the second stage of the algorithm. by Lemma B.1 we have:

$$\mathbb{E}_{\xi^1 \setminus \xi^0}[F(x^2)] - F^\star \leq (1 - \sqrt{\mu_c}/d)^K \cdot 2(F(x^1) - F^\star) + 2\tau(1+c)^2\left(\sqrt{\frac{1}{\mu_c}} + 1\right)v_1^2, \tag{49}$$

and then we take expectation over ξ_K^0:

$$\mathbb{E}_{\xi^1}(F(x^2) - F^\star) \leq (1 - \sqrt{\mu_c}/d)^K \cdot 2\mathbb{E}_{\xi_K^0}(F(x^1) - F^\star) + 2\tau(1+c)^2\left(\sqrt{\frac{1}{\mu_c}} + 1\right)v_1^2, \tag{50}$$

where we set:

$$2\tau(1+c)^2\left(\sqrt{\frac{1}{\mu_c}} + 1\right)v_1^2 \leq \frac{\epsilon_1}{8}, \tag{51}$$

note that $v_1 = \frac{\omega\epsilon_1}{\lambda\rho_t} + \varepsilon$ and $\epsilon_1 > \varepsilon$ it is enough if the following inequality is satisfied:

$$2\tau(1+c)^2\left(\sqrt{\frac{1}{\mu_c}} + 1\right)\left(\frac{\omega\epsilon_1}{\lambda\rho_t} + \epsilon_1\right)^2 \leq \frac{\epsilon_1}{8} \tag{52}$$

equivalently:

$$2\tau(1+c)^2\left(\sqrt{\frac{1}{\mu_c}} + 1\right)\left(\frac{\omega}{\lambda\rho} + 1\right)^2 \frac{4d^2\mathcal{D}(x^0, x^\star)}{(2d + K_0)^2} \leq \frac{\epsilon_1}{8}, \tag{53}$$

hence it is enough to set:

$$K_0 = \left\lceil 8d(1 + \frac{\omega}{\lambda\rho})\sqrt{(\sqrt{\frac{1}{\mu_c}} + 1)\tau(1+c)^2\mathcal{D}(x^0, x^\star)} \right\rceil \tag{54}$$

Then if we choose:

$$K = \left\lceil \frac{\log 16}{\log \frac{1}{(1 - \sqrt{\mu_c}/d)}} \right\rceil, \tag{55}$$

we can ensure that:

$$\mathbb{E}_{\xi^1}(F(x^2) - F^\star) \leq \frac{\epsilon_1}{8} + \frac{\epsilon_1}{8} = \frac{\epsilon_1}{4} = \epsilon_2. \tag{56}$$

Induction part 2: For $t+1$-th outer iteration, by induction hypothesis on t-th outer iteration which reads: $\mathbb{E}_{\xi^{t-1}}F(x^t) - F^\star \leq \frac{\epsilon_{t-1}}{4} = \epsilon_t$, we can write:

$$\mathbb{E}_{\xi^t \setminus \xi^{t-1}}(F(x^{t+1}) - F^\star) \leq (1 - \frac{\sqrt{\mu_c}}{d})^K \cdot 2(F(x^t) - F^\star) + 2\tau(1+c)^2\left(\sqrt{\frac{1}{\mu_c}} + 1\right)v_t^2, \tag{57}$$

with probability at least $1 - \frac{\rho_t}{2}$. Then we take expectation over ξ_K^{t-1}:

$$\mathbb{E}_{\xi^t}(F(x^{t+1}) - F^\star) \leq (1 - \frac{\sqrt{\mu_c}}{d})^K \cdot 2\mathbb{E}_{\xi^{t-1}}(F(x^t) - F^\star) + 2\tau(1+c)^2\left(\sqrt{\frac{1}{\mu_c}} + 1\right)v_t^2, \tag{58}$$

where we need:

$$2\tau(1+c)^2\left(\sqrt{\frac{1}{\mu_c}} + 1\right)v_t^2 \leq \frac{\epsilon_t}{8}, \tag{59}$$

since we have chosen that $\rho_t = \frac{1}{2}\rho_{t-1}$ and $\epsilon_t = \frac{1}{4}\epsilon_{t-1}$, then $v_t \leq v_{t-1} \leq .. \leq v_1$, the above inequality is satisfied by our choice of K_0.

Again if we choose:

$$K = \left\lceil \frac{\log(16)}{\log \frac{1}{(1-\sqrt{\mu_c}/d)}} \right\rceil, \tag{60}$$

we can ensure that:

$$\mathbb{E}_{\xi^t}(F(x^{t+1}) - F^\star) \leq \frac{\epsilon_t}{8} + \frac{\epsilon_t}{8} = \frac{\epsilon_t}{4} = \epsilon_{t+1}. \tag{61}$$

with probability at least $1 - \sum_{i=1}^{t} \frac{\rho_i}{2} \geq 1 - \rho$, , for $\delta \geq \varepsilon$. Hence we finish the induction and the proof of Theorem 2.8.

In summary for Two-Stage APCG if we choose $K := \left\lceil \frac{\log 16}{\log \frac{1}{(1-\sqrt{\mu_c}/d)}} \right\rceil$, if the number of coordinate gradient oracle calls N satisfies:

$$N := tK + K_0 \geq \left\lceil \frac{\log 16}{\log \frac{1}{(1-\sqrt{\mu_c}/d)}} \right\rceil \log_4(\frac{[F(x^1) - F^\star]}{\delta}) + K_0, \tag{62}$$

we have $\mathbb{E}_{\xi^{t-1}} F(x^t) - F^\star \leq \delta$, which is claimed in Corollary 2.9. □

C Adaptive Two-Stage APCG (Option 2) via a simple heuristic procedure for estimating μ_c

In this appendix we provide a heuristic approach of estimating μ_c for Two-Stage APCG (Option 2).

We describe the intuition of this procedure. First we observe that for $F(x^t) - F^\star < 1$, the convergence speed of the second stage algorithm reads:

$$\begin{aligned}
& \mathbb{E}_{\xi_K^t \setminus \xi_K^{t-1}}[F(x^{t+1})] - F^\star \\
\leq\ & \left(1 - \frac{\sqrt{\mu_c}}{d}\right)^K 2\left[F(x^t) - F^\star\right] \\
& + 2\tau(1+c)^2 \left(\sqrt{\frac{1}{\mu_c}} + 1\right) v_t^2 \\
\approx\ & \left(1 - \frac{\sqrt{\mu_c}}{d}\right)^K 2\left[F(x^t) - F^\star\right] + o\left[F(x^t) - F^\star\right] \\
\approx\ & \left(1 - \frac{\sqrt{\mu_c}}{d}\right)^K 2\left[F(x^t) - F^\star\right].
\end{aligned}$$

Directly using this relationship to check the convergence speed is impossible because $F^\star$ is unknown beforehand in general, but it has been shown in (Fercoq and Qu, 2017, Prop. 4) that $F(x) - F^\star$ can be lower bounded as:

$$F(x) - F^\star \geq O(\|\mathcal{G}(x) - x\|_2^2), \tag{63}$$

where $T(x)$ is the composite gradient map:

$$\mathcal{G}(x) = \arg\min_{q \in \mathbb{R}^d} \frac{d \max_i L_i}{2} \|x - q\|_2^2 + \langle \nabla f(x), q - x \rangle + \lambda g(q), \tag{64}$$

and meanwhile there is also upper bound : $F(x) - F^\star \leq O(\|\mathcal{G}(x) - x\|_2^2)$.

Hence our heuristic procedure's checking condition is built based on a simplified version of the above relationship by dropping the expectation:

$$\|\mathcal{G}(x^{t+1}) - x^{t+1}\|_2^2 \lesssim C \left(1 - \frac{\sqrt{\mu_c}}{d}\right)^K \|\mathcal{G}(x^t) - x^t\|_2^2 \tag{65}$$

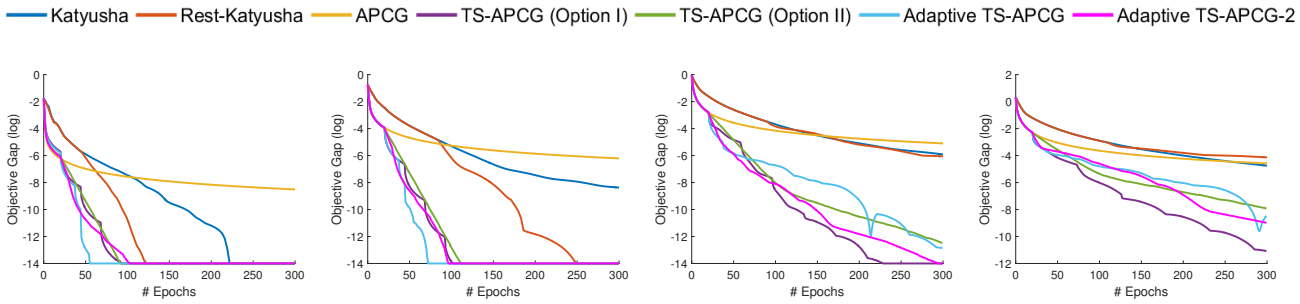
where the variable C represent the strictness of the condition. In the adaptive algorithm we check the condition (65) every $K = \left\lceil \frac{\log 16}{\log \frac{1}{(1-\sqrt{\mu_t}/d)}} \right\rceil$ of iterations where μ_t is the current estimate of μ_c, if it is violated we suspect that our estimation of μ_c is too large and hence we shrink it by a factor of 2 and then restart the second stage algorithm, otherwise we double the estimate to ensure that we choose the estimation of μ_c as aggressive as possible. If we observe that the algorithm is shrinking the μ_c for a number of times in a row, we suspect that the algorithm's checking condition is too strict and hence we double C to relax the condition.

Algorithm 3 Adaptive Two-Stage APCG - 2 (x^0, μ_1, K_0, C, T)

$x^1 = \text{APCG}_0\ (x^0, K_0)$
Calculate the composite gradient map $\mathcal{G}(x^1)$ by eq:(64).
for $t = 1, \ldots, T$ **do**
  $x^{t+1} = \text{APCG}\ (x^t, K, \mu_t)$
  ——Track the convergence speed :
    Calculate $\mathcal{G}(x^{t+1})$ by eq:(64)
  —— Update the estimate of RSC
    if $\|\mathcal{G}(x^{t+1}) - x^{t+1}\|_2^2 \lesssim C\left(1 - \frac{\sqrt{\mu_t}}{d}\right)^K \|\mathcal{G}(x^t) - x^t\|_2^2$
      then $\mu_{t+1} \leftarrow 2\mu_t$, **else** $\mu_{t+1} \leftarrow \mu_t/2$.
  ——Adaptively tune the restart period :
    $K = \left\lceil \frac{\log 16}{\log \frac{1}{(1-\sqrt{\mu_t}/d)}} \right\rceil$
    if $\mu_{t+1} \leq 2^{-5}\mu_{t-4}$ **then** $C \leftarrow 2C$
    if $\mu_{t+1} \geq 2^5\mu_{t-4}$ **then** $C \leftarrow \max(1, \frac{C}{2})$
  end if
end for

C.1 Additional Experimental Results for the Adaptive Variant of Option 2

In this section we present an additional lasso experimental result with the Adaptive Two-Stage APCG-2 algorithm (pink lines) on Madelon dataset with extra 3500 random features. We set the initial guess of the RSC parameter $\mu_1 = 0.1$, the same as the adaptive variant of Option 1 described in the main text. We see that the adaptive variant of two-stage APCG's option 2 also can achieve comparable results without the explicit knowledge of μ_c but estimate it on the fly:



(b) $\lambda = 1 \times 10^{-3}$, $\|x^\star\|_0 = 126$(c) $\lambda = 5 \times 10^{-4}$, $\|x^\star\|_0 = 618$(d) $\lambda = 2 \times 10^{-4}$, $\|x^\star\|_0 = 1250$(e) $\lambda = 1 \times 10^{-4}$, $\|x^\star\|_0 = 1594$

Figure 5: Lasso regression on Madelon dataset with additional random features ($A \in \mathbb{R}^{2000 \times 4000}$)